

# Trinity and NERSC-8 Computing Platforms: Draft Technical Requirements

---

<b>1 INTRODUCTION</b>	<b>3</b>
1.1 TRINITY	3
1.2 NERSC-8	4
1.3 HIGH-LEVEL SCHEDULE	5
<b>2 MANDATORY DESIGN REQUIREMENTS</b>	<b>5</b>
<b>3 TARGET DESIGN REQUIREMENTS</b>	<b>6</b>
3.1 SCALABILITY	6
3.2 SYSTEM SOFTWARE AND RUNTIME	7
3.3 SOFTWARE TOOLS AND PROGRAMMING ENVIRONMENT	8
3.4 PARALLEL FILE SYSTEM	10
3.5 APPLICATION PERFORMANCE REQUIREMENTS	11
3.6 RESILIENCE, RELIABILITY & AVAILABILITY	13
3.7 SYSTEM OPERATIONS	14
3.8 BUILDABLE SOURCE CODE	15
3.9 FACILITIES AND SITE INTEGRATION	15
3.10 TARGET SYSTEM CONFIGURATIONS	18
<b>4 OPTIONS</b>	<b>19</b>
4.1 VISUALIZATION AND DATA ANALYSIS	19
4.2 BURST BUFFER	20
4.3 ADVANCED POWER MANAGEMENT	21
4.4 APPLICATION TRANSITION SUPPORT	22
4.5 EARLY ACCESS DEVELOPMENT SYSTEM	22
4.6 TEST SYSTEMS	23
4.7 ON SITE SYSTEM AND APPLICATION SOFTWARE ANALYSTS	23
4.8 ALTERNATIVE PROPOSALS	23
4.9 ADDITIONAL SYSTEM OPTIONS	23
<b>5 DELIVERY AND ACCEPTANCE REQUIREMENTS</b>	<b>24</b>
5.1 PRE-DELIVERY TESTING	24

<b>5.2</b>	<b>SITE INTEGRATION AND POST-DELIVERY TESTING</b>	<b>25</b>
<b>5.3</b>	<b>ACCEPTANCE TESTING</b>	<b>25</b>
<b>6</b>	<b>TECHNICAL SERVICES, DOCUMENTATION AND TRAINING</b>	<b>25</b>
<b>7</b>	<b>VENDOR CAPABILITIES &amp; RISK MANAGEMENT</b>	<b>25</b>
<b>8</b>	<b>GLOSSARY</b>	<b>25</b>
<b>9</b>	<b>REFERENCES</b>	<b>27</b>

Draft

# 1 Introduction

The National Energy Research Scientific Computing (NERSC) Center and the Alliance for Computing at Extreme Scale (ACES), a collaboration between Los Alamos National Laboratory and Sandia National Laboratory are partnering to release a joint Request for Proposal (RFP) for two next generation systems, Trinity and NERSC-8, to be delivered in the 2015 time frame. The intention is to choose a single vendor to deliver two systems of similar technology. The technical specifications in this document describe joint requirements everywhere except for the tables in Section 3 that describe requirements specific to the Trinity and NERSC-8 systems.

Trinity and NERSC-8 each have maximum funding limits over their system lives, to include all design and development, maintenance, support and analysts.

The Offeror must respond with a configuration and price for both systems.

## 1.1 Trinity

The DOE NNSA ASC Program requires a computing system be deployed in 2015 to support the Stockpile Stewardship Program. In the 2015 timeframe, the current ASC systems will be nearing the end of their useful lifetime. Trinity, the proposed Advanced Technology System(ATS), provides a replacement, tri-lab computing resource for existing simulation codes and provides a larger resource for ever-increasing computing requirements to support the weapons program. The Trinity system, to be sited at Los Alamos, NM, is projected to provide a large portion of the ATS resources for the NNSA ASC tri-lab simulation community: Los Alamos National Laboratory (LANL), Sandia National Laboratories (SNL), and Lawrence Livermore National Laboratory (LLNL), during the 2016-2020 timeframe.

In order to fulfill its mission, the NNSA Stockpile Stewardship Program requires higher performance computational resources than are currently available within the Nuclear Security Enterprise (NSE). These capabilities are required for supporting stockpile stewardship certification and assessments to ensure that the nation's nuclear stockpile is safe, reliable, and secure.

The ASC Program is faced with significant challenges by the on-going technology revolution. It must continue to meet the mission needs of the current applications but also must adapt to radical change in technology in order to continue running the most demanding applications in the future. The ASC Program recognizes that the simulation environment of the future will be transformed with new computing architectures and new programming models that will take advantage of the new architectures. Within this context, ASC recognizes that ASC applications must begin the transition to the new simulation environment or they may become obsolete as a result of not leveraging technology driven by market trends. With this challenge of technology change, it is a major programmatic driver to provide an architecture that keeps ASC moving forward and allows applications to fully explore and exploit upcoming technologies, in addition to meeting NNSA Defense Programs' mission needs. It is possible that major modifications to the ASC simulation tools will be required in order to take full advantage of the new technology, however, existing

codes are expected to run on Trinity. In some cases new applications also may need to be developed. Trinity is expected to help technology development for the ASC Program to meet the requirements of future platforms with greater computational performance or capability. Trinity will serve as a technology path for future ASC systems in the next decade.

To directly support the ASC Roadmap, which states that “work in this timeframe will establish the technological foundation to build toward exascale computing environments, which predictive capability may demand,” it is critical for the ASC Program to both explore the rapidly changing technology of future systems and to provide platforms with higher performance and more memory capacity for predictive capability. Therefore, a design goal of Trinity is to achieve a balance between usability of current NNSA ASC simulation codes and adaptation to new computing technologies.

## **1.2 NERSC-8**

The U.S. Department of Energy (DOE) Office of Science (SC) requires a high performance production computing system in the 2015/2016 timeframe to support the rapidly increasing computational demands of the entire spectrum of DOE SC computational research. The system needs to provide a significant upgrade in computational capabilities, with a target increase between 10-30 times the sustained performance over the NERSC-6 Hopper system.

In addition to increasing the computational capability available to DOE computational scientists, the system also needs to be a platform that will begin to transition DOE scientific applications to more energy-efficient, many-core architectures. This need is closely aligned with the US Department of Energy’s 2011 strategic plan, which states an imperative to continue to advance the frontiers of energy-efficient computing and supercomputing to enable greater computational capacity with lower energy needs. Energy-efficient computing is a cornerstone technology of what has been called exascale computing and represents the only way of continuing NERSC’s historic performance growth in response to science needs.

The NERSC Center supports over 4500 users and 650 applications across a broad range of science disciplines from Chemistry, Material Science, Fusion Energy, Astrophysics, Climate Science and more. The scientific goals driving the need for additional computational capability and capacity are clear. Well-established fields that already rely on large-scale simulation are moving to incorporate additional physical processes and higher resolution. Furthermore new physics are needed to allow more faithful representations of real-world systems, as is the need to model larger systems in more realistic geometries and in finer detail. Additionally, a large and significant portion of the scientific discovery of importance to DOE consists of computational science not performed at the largest scales, but rather, performed using a very large number of individual, mutually-independent compute tasks, either for the purpose of screening or to reduce and/or quantify uncertainty in the results. And finally, the NERSC-8 system must support the rapidly growing computational and storage requirements to support key DOE user facilities and

experiments. For more detail about DOE SC application requirements see: <http://www.nersc.gov/science/requirements-workshops/>

The NERSC-8 system will be housed in the Computational Research and Theory building under construction at Lawrence Berkeley National Laboratory and is expected to run for 4-6 years. The system must integrate into the NERSC environment providing high bandwidth access to existing data stored by continuing research projects.

### 1.3 High-level Schedule

The following is the tentative schedule for the Trinity and NERSC-8 systems.

	Trinity	NERSC 8
RFP Released	Q2CY13	
Contract Awarded	Q3CY13	Q4CY13
On-site System Delivery and Build Complete	Q3CY15	Q4CY15
Acceptance Complete	Q1CY16	Q1CY16

## 2 Mandatory Design Requirements

An Offeror shall address all mandatory requirements and its proposal shall demonstrate how it meets or exceeds each one. A proposal will be deemed non-responsive and will receive no further consideration if any one of the following mandatory requirements is not met.

- 2.1.1 The Offeror shall respond with a single proposal that contains distinct sections showing how and where their proposed Trinity and NERSC-8 systems differ.
- 2.1.2 The Offeror shall provide a detailed architectural description of both the Trinity and NERSC-8 systems. The description shall include: a high-level architectural diagram to include all major components and subsystems; detailed descriptions of all the major architectural hardware components in the system to include: node, cabinet, rack architecture up to the total system, including the high-speed interconnect(s) and network topology; system software components; the storage subsystem and all I/O and file system components; and a proposed floor plan.
- 2.1.3 The Offeror shall describe how the proposed system does or does not fit into their long-term product roadmap and a potential follow-on platform acquisition in the 2019 and beyond timeframe.

### 3 Target Design Requirements

This section contains detailed system design targets and performance features. It is desirable that the Offeror's design meets or exceeds all the features and performance metrics outlined in this section. Failure to meet a given target requirement will NOT make the proposal non-responsive. However, if a target requirement cannot be met it is highly desirable that the Offeror provide a development and deployment plan and schedule to satisfy the requirement.

The Offeror should address all Target Design Requirements and describe how the proposed system meets or does not meet the target design requirements. The Offeror shall also propose any hardware and/or software architectural features that will provide improvements for any aspect of the system. Areas of interest include application performance, resiliency, reliability, power measurement and control, file systems and storage, and system management.

#### 3.1 Scalability

The systems shall be able to support jobs up to the full scale. At any given time, the system workload will include a single job occupying at least one-half (1/2) of the computational partition. As such, the system must scale well to ensure efficient usage.

- 3.1.1 The system shall support running a single application to the full scale.
- 3.1.2 The system shall support an efficient, scalable mechanism to launch applications at sizes up to full scale in under 30 seconds. Offerors shall describe the factors (such as executable size) that affect application launch time.
- 3.1.3 The system shall support hundreds of concurrent users and tens of thousands of concurrent batch jobs. The Offeror shall describe and provide details on the method to support this requirement.
- 3.1.4 The Offeror shall describe all areas of the system in which node-level resource usage (hardware and software) increases in size as a job scales to larger sizes.
- 3.1.5 The system's high-speed interconnect shall support high bandwidth, low latency, high throughput, and independent progress. The Offeror shall describe the high-speed interconnect in detail, including any mechanisms for adapting to heavy loads or inoperable links.
- 3.1.6 The system shall utilize an optimized job placement algorithm to reduce job runtime, lower variability, minimize latency, etc. The Offeror shall describe in detail how the algorithm is optimized to the system architecture.

- 3.1.7 The system shall provide an application programming interface to allow applications access to the physical to logical mapping information of the job's node allocation.
- 3.1.8 The Offeror shall describe how the system software solution provides a low jitter environment for applications and shall provide an estimate of a compute node OS's noise profile, both while idle and while running a non-trivial MPI application. If core specialization is used, describe the system software activity that remains on the application cores.
- 3.1.9 The system shall provide correct and consistent runtimes. An application's runtime (i.e. wall clock time) shall not change by more than 3% from run-to-run in dedicated mode and 5% in production mode.

## **3.2 System Software and Runtime**

The Offeror shall propose a well-integrated and supported system software environment. The overall imperative is to provide users with a productive, high-performing, and reliable system software environment by which to use the system.

- 3.2.1 The system shall include (i) a full-featured Unix-like operating system (OS) environment on all user visible service partitions, (e.g. login nodes, service nodes) and for the system management services and (ii) a compute partition OS that provides an efficient execution environment of the applications running at full-system scale. The Offeror shall describe in detail the overall system software architecture.
- 3.2.2 The full-featured Unix-like operating system for the service nodes and for the system management workstations shall provide at a minimum the following security features: ssh version 2, Unix/Linux user and group permissions, access control lists, kernel-level firewall capabilities, logging, and auditing. The Offeror shall describe the security capabilities of the full-featured operating system.
- 3.2.3 The compute partition OS shall provide a trusted, hardware-protected supervisory mode to implement security features. The supervisor/kernel shall provide authoritative user identification, ensure that user access controls are in place, employ the principle of least privilege, and interoperate with the same features on the service nodes and management workstation(s). Logging and auditing features supported by the compute node operating system shall have the capability to be enabled, disabled and custom configured to site preferences. The Offeror shall provide details of the security features of their compute node operating system(s).
- 3.2.4 The system shall provide efficient support for dynamic loading of shared objects, including `dlopen()`, and shall support applications using these techniques at the full scale of the system.

- 3.2.5 The system shall provide efficient, secure interprocess communication that allows cooperating applications running anywhere on the high-speed network to inter-communicate (e.g. the compute partition, the service partition, or both). The provided mechanism shall be as close to the underlying network stack as possible. The security model shall allow applications and users to set access controls based on authenticated or trusted values for process identifier and user identifier.
- 3.2.6 The Offeror shall provide a documented and efficient application programming interface (API) for the native network layer(s) of the high-speed network software stack.
- 3.2.7 The system shall provide resource management functionality including checkpoint-restart, job migration, backfill, targeting of specified resources, advance and persistent reservations, job preemption, job accounting, and architecture-aware job placement. The Offeror may propose multiple options for a vendor-supported resource manager, one of which shall be compatible with Adaptive Computing's Moab product.
- 3.2.8 The resource manager shall accept jobs submitted via the Globus tool kit.

### **3.3 Software Tools and Programming Environment**

The primary programming model used by application scientists running on existing ASC and NERSC systems is MPI. The scientific application community recognizes that in order to achieve application performance on future, more energy efficient architectures, application developers will need to transition to an MPI+X programming model where MPI continues to serve as the programming model for inter-node communication and X provides for finer-grain, on-node parallelism. To support legacy applications the Offeror's proposed system shall continue to support the MPI programming model.

- 3.3.1 The system shall support the Message Passing Interface (MPI) 3 standard specification. The Offeror shall provide a detailed description of the MPI implementation, including version, support for features such as accelerated collectives, and describe any limitations relative to the MPI 3 standard.
- 3.3.2 The Offeror shall describe at what level the system can be utilized by MPI-only applications.
- 3.3.3 The Offeror shall provide optimized implementations for key inter-node and intra-node MPI collective operations, including MPI\_BARRIER, MPI\_ALLREDUCE and MPI\_ALLGATHER.

- 3.3.4 The Offeror shall provide an efficient implementation of MPI\_THREAD\_MULTIPLE. Bandwidth, latency and message throughput measurements using the MPI\_THREAD\_MULTIPLE thread support level shall have no more than a 10% performance degradation when compared to using the MPI\_THREAD\_SINGLE support level.
- 3.3.5 The Offeror shall describe in detail all programming APIs, languages, compiler extensions, etc. other than MPI (e.g. OpenMP, OpenACC, CUDA, etc.) that will be supported. Describe the advantages and disadvantages of each node level programming API from a programming and performance perspective. In addition, describe any interoperability limitations (e.g. thread interoperability).
- 3.3.6 The system shall enable applications to control task and memory placement within a node for efficient performance. The Offeror shall provide a detailed description of controls provided and any limitations that may exist.
- 3.3.7 The system shall support the languages C, C++, Fortran 77, Fortran 2008, and Python on the compute partition. It is highly desirable to provide multiple compilation environments. The Offeror shall list all languages and compile environments, including version numbers.
- 3.3.8 The system shall support partitioned global address space (PGAS) languages and memory communications. Describe system hardware and programming environment software for exploiting partitioned global address space (PGAS) capabilities.
- 3.3.9 The system shall include optimized versions of libm, libgsl, FFTW, BLAS1-3, LAPACK/ScaLAPACK, HDF5 and netCDF. The Offeror shall describe all optimized libraries that will be supported.
- 3.3.10 The system shall include a comprehensive software development environment with configuration and source code management tools.
- 3.3.11 The system shall provide an interactive debugger with an X11-based graphical user interface. The debugger shall provide a single-point of control that can debug applications using every level of parallelism and programming environment provided by the system.
- 3.3.12 The system shall provide a suite of tools for detailed performance analysis and profiling of user applications. The tools shall support all levels of parallelism and programming environment provided in the system. The tools shall be capable of supporting a single job at the full scale of the system.

- 3.3.13 The system shall provide event-tracing tools. Event Tracing of interest include: Message-Passing Event Tracing, I/O Event Tracing, Floating Point Exception Tracing, and Lightweight Message-Passing Profiling. The event-tracing tool API shall provide functions to activate and deactivate event monitoring during execution from within a process.
- 3.3.14 The system shall provide stack-tracing tools. The tool set shall include a source-level stack trace back, including an API that allows a running process or thread to query its current stack trace.

### **3.4 Parallel File System**

- 3.4.1 The system shall include a closely coupled parallel file system (PFS) that presents a global, consistent name space to the platform. The Offeror shall provide a detailed description of the PFS implementation including
- File system architecture and proposed implementation,
  - Expected scaling characteristics,
  - Management, diagnostic, deployment, security and configuration tools,
  - Externalized error and diagnostic information.
- 3.4.2 The parallel file system shall exhibit zero data corruption and zero data loss over the life of the platform. Describe in detail how the I/O solution will achieve these traits including how you address silent data corruption errors (e.g. high fly writes, short writes, misdirected I/Os). Provide estimates of failure rates. Reliability of the PFS will be assessed as part of the overall system reliability metrics.
- 3.4.3 The PFS shall achieve the target bandwidth, as specified in Table 4, using any arbitrary collection of compute nodes starting at 10% of compute nodes up to the full scale of the system when the PFS is up to 70% full. Describe the extent to which performance differs with the size of transfers or the number of files being read/written.
- 3.4.4 The PFS shall provide a robust, interactive environment for users. The time required to perform an insert, delete, enumerate, and retrieve file system object status within a single directory on login or file transfer service nodes shall be prompt and not be substantially impacted by unrelated applications running on the compute partition. Given a single directory with 1 million files, describe how long the following metadata operations will take on the proposed file system
- Insert one million objects,
  - Delete one million objects,
  - Enumerate and retrieve one million objects.

- 3.4.5 The system shall provide POSIX I/O and MPI-2 I/O functionality that is tightly integrated with file system software to provide high performance small and large block I/O and for single and shared files. MPI I/O shared file performance shall achieve 80% of POSIX I/O performance using a single file per processor at full system bandwidth.
- 3.4.6 The PFS system shall support access to external clients with the same functionality as internal clients.
- 3.4.7 The Offeror shall describe expected PFS maintenance procedures and their impacts on the PFS performance under normal load and other routine operations including purging, file system health monitoring, performance statistics, problem alerts, diagnosis and repair, and reconstruction after a drive replacement, including reconstruction time.

### **3.5 Application Performance Requirements**

Assuring that real applications perform well on the Trinity and NERSC-8 platforms is key to the success of the systems. Because the full applications are large, often with millions of lines of code, are typically written using MPI only, and in some cases are export controlled, NERSC and ACES have put together a suite of ‘mini-apps’ and micro-benchmarks for RFP evaluation and system acceptance. The mini-apps listed in Table 1 are representative of the two workloads, but are smaller than full applications. All performance results reported in the Offeror’s proposal will become requirements at acceptance time. The performance of mini-apps listed in Table 1 will be evaluated at time of selection and acceptance. The performance of the micro-benchmarks listed in Table 2 will be evaluated as indicated in the table.

For mini-applications and the ASC code suite, source code and example run time rules will be provided to the Offeror, but may require compliance with export control laws and/or no cost licensing agreements.

- 3.5.1 The Offeror shall provide performance results (actual, predicted and/or extrapolated) for the proposed systems for all mini-applications and benchmarks listed in Table 1 and Table 2. If predicted or extrapolated results are provided, explain the methodology used.

The Offeror shall report all benchmark results in the accompanying “Benchmark Run Rules and Results” worksheet. The draft benchmarks, input data sets and run rules can be found on this website: <https://www.nersc.gov/systems/trinity-nersc-8-rfp/>.

- 3.5.2 Offeror shall provide licenses for the delivered system for all compilers used to achieve benchmark performance.

- 3.5.3 The Offeror’s proposal shall state a minimum Sustained System Performance (SSP) for both the Trinity and NERSC-8 systems as measured by the SSP metric for the mini-app code suite defined in Table 1. The target SSP is stated in table Table 4 as an increase over the Hopper system. The baseline Hopper SSP will be stated in the final RFP technical requirements. Background to the SSP metric is defined here: <http://www.nersc.gov/research-and-development/performance-and-monitoring-tools/sustained-system-performance-ssp-benchmark/>.
- 3.5.4 In addition to using the mini-apps, for final acceptance of the Trinity system, an ASC Simulation Code Suite will be used to judge the performance. The Trinity system shall achieve, on average, at least 8 times (8x) capability improvement over the ASC Cielo platform [1].
- Cielo results shall be collected at a nominal scale of 8,192 nodes and will be provided to the Offeror.
  - Trinity results shall be collected using at least two-thirds (2/3) of the Trinity system.
  - Capability improvement is defined as the product of an increase in problem size and an application specific runtime speedup factor. For example, if the problem size is 8 times larger and the runtime speedup is 1.2, the capability improvement is 9.6 (9.6x).
  - The code suite will include at least two (2) to four (4) ASC applications. Source code and example run time rules will be provided to the Offeror, but may require compliance with export control laws and no cost licensing agreements.

Table 1. Mini Applications

Mini-App Name	Description
miniFE	Unstructured implicit finite element
miniGhost	Finite difference stencil
miniContact	Contact search
AMG	Algebraic Multi-Grid linear system solver for unstructured mesh physics packages
UMT	Unstructured-Mesh deterministic radiation Transport
miniPartiSN	Structured Particle Transport Surrogate
miniDFT	Density Functional Theory (DFT)
GTC	Particle-in-cell magnetic fusion

MILC	Lattice Quantum Chromodynamics (QCD). Sparse matrix inversion, CG
------	---

Table 2. Micro-benchmarks

Benchmark	Selection	Acceptance	Section
STREAM	X	X	N/A
Pynamic		X	3.2.4
Ziatest		X	3.1.1, 3.1.2
OMB	X	X	3.1.5
SMB	X	X	3.1.5
Metabench		X	3.4.4
IOR		X	3.4.3
UPC NAS		X	3.3.8
PSNAP		X	3.1.8
mpimemu	X	X	3.1.4
TBD: MPI_THREAD_MULTIPLE		X	3.3.4

### 3.6 Resilience, Reliability & Availability

For each metric specified below, the Offeror must describe how they arrived at their estimates. Definitions of terms used in this section can be found in the Glossary.

- 3.6.1 Failure of the system management and/or RAS system(s) shall not cause a system or job interrupt.
- 3.6.2 The minimum Job Mean Time To Interrupt (JMTTI) for a job running on the entire system shall be at least 30 hours.
- 3.6.3 The minimum System Mean Time Between Interrupt (SMTBI) shall be 336 hours.
- 3.6.4 The ratio JMTTI/Delta when using the PFS shall be greater than 30.
- 3.6.5 A job interrupt shall not require a complete resource reallocation.
- 3.6.6 A complete system initialization shall take no more than 30 minutes. The Offeror shall describe the full system initialization sequence and timings.

- 3.6.7 The system shall achieve 99% scheduled availability.
- 3.6.8 The Offeror shall discuss the resilience, reliability and availability mechanisms and capabilities of the proposed system including, but not limited to:
- Any condition or event that can potentially cause a job interrupt,
  - Resiliency features to achieve the availability targets,
  - Single points of failure, hardware or software, and the potential effect on running applications and system availability.
  - How a job maintains its allocation after a node interrupt.

### **3.7 System Operations**

System operation capabilities provide the ability to effectively manage system resources with high utilization and throughput under a workload with a wide range of concurrencies. System management must be an integral part of the overall system. The overall objective is to provide system administrators, security officers, and user-support personnel with a productive and efficient system configuration management and an enhanced diagnostic environment.

- 3.7.1 The Offeror shall deliver scalable integrated system management capabilities that provide human interfaces and APIs for system configuration and its ability to be automated, software management, change management, local site integration, and system configuration backup and recovery.
- 3.7.2 The Offeror shall provide a means for tracking and analyzing all software updates, software and hardware failures, and hardware replacements over the lifetime of the system.
- 3.7.3 The system management capabilities shall provide a single, scalable log analysis capability for all logs originating from any component of the proposed system.
- 3.7.4 Discussion of system configuration management and diagnostic capabilities shall address the following topics:
- Detailed description of the system management support,
  - Any effect or overhead of software management tool components on the CPU or memory available on compute nodes,
  - Release plan, with regression testing and validation for all system related software and security updates,
  - Support for multiple simultaneous or alternative system software configurations, including estimated time and effort required to install both a major and a minor system software update,
  - User activity tracking, such as audit logging and process accounting.

### 3.8 Buildable Source Code

- 3.8.1 Source code, and necessary build environment, shall be provided for all software except for firmware, compilers and third party products.
- 3.8.2 Updates of source code, and any necessary build environment, for all software shall be provided over the life of the contract.

### 3.9 Facilities and Site Integration

- 3.9.1 The system shall use 3-phase 480V AC.
- 3.9.2 All equipment and power control hardware shall be Nationally Recognized Testing Laboratories (NRTL) certified. All equipment shall bear appropriate NRTL labels.
- 3.9.3 The Offeror shall describe the features of the system related to facilities and site integration including:
  - Remote environmental monitoring capabilities of the system and how it would integrate into facility monitoring,
  - Detailed descriptions of power and cooling distributions throughout the system including power consumption for all subsystems, and idle, observed maximum (e.g. HPL), and design limits,
  - OS distributions or other client requirements to support for off-platform access to the parallel file system, such as are used on the LANL File Transfer Agents.
- 3.9.4 All cabling, with the exception of power cabling and water lines, shall be above floor. If it is not possible for a cable to run above the floor, the cable shall be plenum rated and comply with NEC 300.22 and NEC 645.5. All communications cables, wherever installed, shall be source/destination labeled at both ends. All communications cables and fibers over 10 meters in length and installed under the floor shall have a unique serial number and dB loss data (or equivalent) shall be delivered for each cable, if a method to make this measurement exists.

Table 3. Trinity and NERSC-8 Facility Requirements

	Trinity	NERSC-8
Location	Los Alamos National Laboratory, Los Alamos, NM. The system will be housed in the Strategic Computing Complex	National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA.

	Trinity	NERSC-8
	(SCC), Building 2327	The system will be housed in the Computational Theory and Research Facility, now under construction.
Altitude	7,500 feet	200 feet
Seismic	N/A	System to be placed on a seismic isolation floor
Water Cooling	The Offeror will provide water cooling in conformance with ASHRAE Class W3 guidelines (dated 2011). No more than 12 MW. Note: LANL facility will provide inlet water at a nominal 75 ° F. It may go to as low as 60° F based on facility and/or environmental factors.	The system must operate with water at 75°F or below.
Air Cooling	No more than 3MW.	The system must operate with supply air at 76°F or below, with a relative humidity from 30%-70%. No more that 500KW of heat shall be removed by air cooling.
Maximum Power	15MW	6MW
Maximum Power Rate of Change	The aggregate instantaneous platform power should not change by more than 2MW in less than one minute.  The hourly average in platform power should not exceed the 2MW wide power band negotiated at least 2 hours in advance.	

	Trinity	NERSC-8
Floor	42" raised floor	48" raised floor
Ceiling	16 foot ceiling and an 18' 6" ceiling plenum	17'10" ceiling however maximum cabinet height is 9'5"
Footprint	8,000 square feet; 80 feet long and 100 feet deep.	64'x143', or 9152 square feet (inclusive of compute, storage and service aisles). It is preferred that cabinet rows run parallel to the short dimension.
Shipment Dimensions and Weight	No restrictions.	For delivery, system components shall weigh less than 7,000 pounds and shall fit into an elevator whose door is 6ft 6in wide and 9ft 0 in high and whose depth is 8ft 0in.
Floor Loading	The average floor loading (including aisles between rows) shall be no more than 300 pounds per square-foot. A maximum limit of 300 pounds per square foot also applies to all loads during installation.	The floor loading of shall not exceed a uniform load of 500lbs/sf. [Add point loads when available.]
Cabling	All cabling, with the exception of power cabling and water lines, shall be above floor. If it is not possible for a cable to run above the floor, the cable shall be plenum rated and comply with NEC 300.22 and NEC 645.5. All communications cables, wherever installed, shall be source/destination labeled at both ends. All communications cables	All power cabling and water connections shall be below the access floor. All other cabling (e.g. system interconnect) should be above floor and integrated into the system cabinetry. Under floor cables (if unavoidable) shall be plenum rated. All signal cables shall be labeled with a unique serial number at both ends. The dB loss data should be provided for each cable

	Trinity	NERSC-8
	and fibers over 10 meters in length and installed under the floor shall have a unique serial number and dB loss data (or equivalent) shall be delivered for each cable, if a method to make this measurement exists.	over 10 meters.
External network interfaces supported	1G, 10G, 40G, IB	1G, 10G, 40G, IB
Remote access bandwidth requirements	10GB/s over TCP/IP	10 GB/sec over TCP/IP
External client bandwidth to/from local PFS	100 GB/s	70 GB/sec
Local client bandwidth to External File System, e.g. GPFS, NFS	20 GB/s	GPFS mount of externally served file systems at 100 GB/sec

### 3.10 Target System Configurations

Table 4. Target Configuration and Performance Requirements

	Trinity	NERSC-8
Miniapps SSP increase over Hopper[2] system	20-60x	10-30x
Minimum and target aggregate memory on compute partition	2 to 4 PB	1 to 2 PB
Disk Capacity	>30x main memory	>20x main memory
Parallel Debugger Licenses	20 simultaneous users; A single job up to one-fourth (1/4) scale	20 simultaneous users; A single job up to one-fourth (1/4) scale
Compiler licenses for each	20	50

	Trinity	NERSC-8
compiler suite proposed		
Maximum PFS dump time (Delta): Read or write 80% of system memory in	20 minutes	35 minutes
Resource Manager and/or Scheduler Licenses	Licenses must be provided , but have an existing Moab site license	Licenses must be provided

## 4 Options

Burst buffer, visualization, and power management and control capabilities are all important to meeting the needs of DOE Advanced Technology goals.

The technical, business and price information for the options will be evaluated during the selection process. Offerors, therefore, must be as detailed as possible in their response. This section contains options to the base systems. Some options are areas of collaboration between ACES and NERSC with the Offeror to provide functionality that doesn't currently exist, is inadequate in the current marketplace, but could be delivered after acceptance. The quality of the Offeror's response to these options will influence its overall evaluation.

The Offeror shall include relevant technical, business and price information for all options. All options shall be priced separately and shall include delivery, installation, warranty and support for the life of the system. Pricing shall be firm fixed prices.

The Offeror is encouraged to provide proposals for additional areas of collaboration that they feel provide substantial value to the Trinity and NERSC-8 systems: particularly in the areas reliability, resiliency, power usage, application performance and overall productivity of the system.

### 4.1 Visualization and Data Analysis

The system shall be capable of supporting a visualization and data services workload. Data analysis and visualization workloads include

- 1) *Post-processing visualization*: analyzing data from simulations stored on the file system. This will include geometry extraction from data, and may include on-platform rendering. The geometry extraction requires substantial computational resources, good bandwidth into these resources, and good access to the parallel file system. GPUs can benefit rendering operations, but in general visualization requires substantial general purpose processing capabilities.

- 2) *In-situ visualization*: analyzing data in memory, as it is generated from the simulation. This requires use of the main compute resources used in tandem with the simulation.
- 3) *In-transit visualization*: analyzing data off-node as it is generated by simulation.
- 4) *Analysis of large ensembles of data*. This could take place in-transit or in post-processing, and shares requirements with these use cases.

In summary, data analysis workloads require substantial computational resources, and especially must support good bandwidth into these resources. All visualization and data analysis resources shall have equal access to all system resources as the main compute resources, in particular the file system and storage resources. The system shall be capable of providing sufficient bandwidth and low latency for remote display.

4.1.1 The visualization resources shall support the following packages:

- CEI's EnSight (proprietary),
- VisIt (source available), and
- Kitware's ParaView (open source).

These packages use some or all of the following system software capabilities: full support of sockets, dynamic linked libraries, Posix threads, Python scripting, MPI, Mesa3D, NumPy, NetCDF, HDF5 and MPI I/O. ACES and NERSC will provide porting support and any required licensing to the Offeror.

4.1.2 It is desirable that the visualization partition be the same node level architecture as the main compute nodes. If the Offeror determines that the proposed compute node architecture is not consistent with the roadmaps of all three visualization packages listed above, the Offeror shall propose an alternative architecture that is consistent.

4.1.3 The visualization resources shall be tightly integrated into the system and be on the same high-speed network as the main compute resources and have equal access to all other system resources, e.g. file systems and storage.

4.1.4 The size of the visualization partition shall be nominally 5% of the total compute partition.

## 4.2 Burst Buffer

The primary resiliency mechanism used on current large-scale computing systems is application level checkpoint/restart, targeting the parallel file system (PFS, Section 3.4). Traditionally, the PFS is based on disk technology and sizing the PFS for capacity also satisfied the bandwidth requirements. Studies have shown that trend may no longer be the case and in the timeframe of Trinity it will be necessary to size for bandwidth, which is unappealing from an economic perspective. [7][8][9]

Although it may still make economic sense for Trinity and NERSC-8 to be designed using the traditional PFS balance factors, it is highly desirable that Trinity and NERSC-8 be a vehicle to start developing an alternative method for satisfying the PFS demands of the system, in particular designing in a higher performing checkpoint/restart mechanism using a more “tightly coupled”, in-system solution. This document will use the term Burst Buffer for this new subsystem. The underlying goal of the Burst Buffer is to provide a fast storage system so as to improve overall application productivity and resilience compared to a traditional PFS. Although it is predominately for checkpoint/restart, it is desirable for the Burst Buffer subsystem to be a general-purpose solution for other application needs, such as post-processing, in-transit visualization and data analytics.

Functional requirements definition, development and design will be done cooperatively between ACES and NERSC with the Offeror. There will be some functionality required at the time of initial installation and standup, with all functionality being deployed within two (2) years of initial system delivery.

- 4.2.1 The Offeror shall provide a design plan for the Burst Buffer subsystem taking into account the design guidelines found in this section.
- 4.2.2 The primary usage model for the Burst Buffer shall be application level checkpoint/restart.
- 4.2.3 The Burst Buffer shall provide a minimum of three (3) times the aggregate main memory capacity of the systems compute partition.
- 4.2.4 The ratio  $JMTTI/\Delta$  when using the Burst Buffer for check-pointing shall be greater than 200.
- 4.2.5 The Burst Buffer shall be designed in conjunction with the PFS. However, the PFS shall be capable of operating without the presence of the Burst Buffer and each shall have separate failure domains.
- 4.2.6 The Burst Buffer shall have the necessary functionality to support data analytics use cases, such as post processing and in-transit visualization.
- 4.2.7 The Burst Buffer shall be capable of being partitioned and allocated on a per job basis. This will require tight integration with the system batch scheduler and job launch mechanism.
- 4.2.8 Reliability of the Burst Buffer will be assessed as part of the overall system reliability metrics.

### **4.3 Advanced Power Management**

Power measurement and control capabilities (hardware and software tools and application programming interfaces (APIs)) are necessary to meet the needs of future supercomputing energy and power constraints. It is extremely important that the Trinity and NERSC-8 projects utilize early capabilities in this area and start

defining and developing advanced capabilities and integrating them into a user friendly, production environment.

Some functionality will be required at time of initial acceptance. Both initial and advanced capabilities will be defined cooperatively between ACES and NERSC and the successful Offeror. All functionality will be required to be deployed within two (2) years of initial system delivery. The Offeror shall:

- 4.3.1 Describe all power related measurement and control features, capabilities and limitations (hardware and software) of the system including, but not limited to, any tools, system software features and APIs that will be made available at initial acceptance.
- 4.3.2 Describe all power related measurement and control capabilities projected on the Offeror's road map. ACES, NERSC, and the successful Offeror will work cooperatively to define a set of capabilities that will be delivered beyond initial acceptance.
- 4.3.3 Describe all power related measurement and control capabilities (hardware and software) that would necessitate hardware upgrade or replacement.

#### **4.4 Application Transition Support**

- 4.4.1 The Offeror shall propose a vehicle (e.g. a center of excellence) for supporting the transition of major applications to the Trinity and NERSC-8 systems. Support will be required from the successful Offeror and all of its key advanced technology providers, e.g. processor vendors. Activities will require the support of experts in the areas of application porting and performance optimization. Support is required from the date of subcontract execution through two (2) years after final acceptance.

#### **4.5 Early Access Development System**

To allow for early and/or accelerated development of applications or development of functionality required as a part of the statement of work, the Offeror shall propose options for early access development systems. These can be in support of the baseline requirements or any proposed options. The Early Access system shall be delivered before Q3CY2014.

- 4.5.1 The Offeror shall propose an Early Access Development System. The primary purpose is to expose the application to the same programming environment as will be found on the final system. It is acceptable for the early access system to NOT use the final processor, node or high-speed interconnect architectures. However, the programming and runtime environment must be sufficiently similar that a port to the final system is trivial. The early access system shall contain similar functionality of the final system, including file systems, but scaled down to the appropriate

configuration. The Offeror shall propose an option for the following configurations based on the size of the final Trinity system.

4.5.1.1 2% of Trinity's compute partition.

4.5.1.2 5% of Trinity's compute partition.

4.5.1.3 10% of Trinity's compute partition.

4.5.2 If applicable, the Offeror shall propose development test bed systems that will reduce risk and aid the development of any advanced functionality that is exercised as a part of the statement of work, e.g. Burst Buffer, power management, etc.

## **4.6 Test Systems**

The Offeror shall propose the following test systems. The systems shall contain all the functionality of the main system, including file systems, but scaled down to the appropriate configuration. Multiple test systems may be awarded.

4.6.1 The Offeror shall propose an Application Regression test system, which shall contain at least 200 compute nodes.

4.6.2 The Offeror shall propose a System Development test system, which shall contain at least 50 compute nodes.

## **4.7 On Site System and Application Software Analysts**

4.7.1 The Offeror shall propose and separately price up to two (2) System Software Analysts and up to two (2) Applications Software Analysts for each site. For Trinity, these positions require a DOE Q-clearance for access.

## **4.8 Alternative Proposals**

4.8.1 Alternative solutions that the Offeror feels would provide value, additional performance for the system and/or reduce risk. The Offeror should note that any Alternative Proposal must still meet all Mandatory Requirements.

## **4.9 Additional System Options**

It is anticipated that NERSC and ACES will have future requirements for system upgrades and/or additional quantities based on the configurations proposed for Trinity and NERSC 8, respectively. To address these potential requirements, the Offeror shall propose and separately price options for system upgrades as indicated in Section 4.9.1, and additional systems as indicated in Section 4.9.2. Since these options will be based upon the Trinity and NERSC 8 technical solutions, the Offeror's proposal need not repeat any previously proffered technical solution; however, the Offeror shall address any technical challenges foreseen with regard to scaling and any other production issues.

- 4.9.1 Upgrade the Trinity and NERSC 8 configurations by the following fractions of the proposed systems:
- 4.9.1.1 25%
  - 4.9.1.2 50%
  - 4.9.1.3 100%
  - 4.9.1.4 200%
- 4.9.2 Provide additional quantities of the Trinity and NERSC 8 configurations at the following fractions of the proposed systems:
- 4.9.2.1 50%
  - 4.9.2.2 100%
  - 4.9.2.3 200%
  - 4.9.2.4 500%
- 4.9.3 Offeror shall propose double the main memory capacity per node.
- 4.9.4 Offeror shall propose additional parallel file system storage in 5PB increments.
- 4.9.5 Other product or service options that Offeror believes would materially benefit Trinity and/or NERSC-8.

## **5 Delivery and Acceptance Requirements**

Testing of the system shall proceed in three steps: pre-delivery, post-delivery and acceptance. Each step is intended to validate the system and feeds into subsequent activities.

### **5.1 Pre-delivery Testing**

ACES, NERSC and vendor staff shall perform pre-delivery testing at the factory on the hardware to be delivered. Any limitations for performing the pre-delivery testing need to be identified including scale and licensing limitations. During pre-delivery testing, the successful Offeror (Subcontractor) shall:

- Demonstrate RAS capabilities and robustness, using simple fault injection techniques such as disconnecting cables, powering down subsystems, or installing known bad parts.
- Demonstrate functional capabilities on each segment of the system built, including the capability to build applications, schedule jobs, and run them using the customer-provided testing framework. The root cause of any application failure must be identified.

- Provide a file system sufficiently provisioned to support the suite of tests.
- Instill confidence in the ability to conform to the statement of work.
- Provide onsite and remote access for ACES and NERSC staff to monitor testing and analyze results.

## **5.2 Site Integration and Post-delivery Testing**

ACES, NERSC and vendor staff shall perform site integration and post-delivery testing on the fully delivered system. Limitations may exist for vendor access to the onsite system.

- During post-delivery testing, the pre-delivery tests shall be run on the full system installation.
- Where applicable, tests shall be run at full scale.

## **5.3 Acceptance Testing**

ACES, NERSC and vendor staff shall perform onsite acceptance testing on the fully installed system. Limitations may exist for vendor access to the onsite system.

- The Subcontractor shall demonstrate that the delivered systems conform to the subcontract's Statement of Work. A sample test plan is provided as a basis for responding to this RFP.

## **6 Technical Services, Documentation and Training**

Technical services, documentation, and training shall provide the operators, system administrators and users of the proposed solution with the information needed to effectively operate, configure and use the platform. ACES and NERSC may, at their option, make audio and video recordings of presentations from Subcontractor's speakers at public events targeted at the ACES and NERSC user communities (e.g., user training events, CoE, Best Practices discussions). Subcontractor grants ACES and NERSC use and distribution rights of vendor provided documentation, session materials and recorded media.

6.1.1 Offeror's proposal shall provide narrative to describe the following training and documentation requirements:

- Classroom training
- Onsite training
- Online documentation

## **7 Vendor Capabilities & Risk Management**

Offeror's proposal shall:

- 7.1.1 Provide a risk management strategy for the proposed system in case of technology problems or scheduling delays that affect availability or achievement of performance targets in the proposed timeframe. Describe the impact of substitute technologies on the overall architecture and performance of the system as described in section 2.1.2. In particular, the Offeror shall address the three technology areas listed below.
- Processor
  - Memory
  - High-speed interconnect
- 7.1.2 Identify any other high-risk areas and accompanying mitigation strategies for the proposed system.
- 7.1.3 Provide a clear plan for effectively responding to software and hardware defects and system outages at each severity level and document how problems or defects will be escalated.
- 7.1.4 Provide a roadmap showing how the response to this procurement aligns with their plans for Exascale Computing.
- 7.1.5 Discuss additional capabilities including the Offeror's:
- Ability to produce and maintain the proposed system for the life of the platform,
  - Ability to achieve specific quality assurance, reliability and availability goals,
  - In-house testing and problem diagnosis capability, including hardware resources at appropriate scale.

## 8 Glossary

- a) Job Interrupt: Any system event that causes a job failure. The ability for the system to recover does not negate a job interrupt and ability to recover will not be considered in the JMTTI calculation.
- b) Job Mean Time to Interrupt (JMTTI): Average time between job interrupts over a given time interval.
- c) System Interrupt: Any system event, or accumulation of system events over time, resulting in more than 1% of the compute resource being unavailable at any given time. Loss of access to any dependent subsystem, e.g. parallel file-system or service partition resource, will also incur a system interrupt.
- d) System Mean Time Between Interrupt (SMTBI): Average time between system interrupts over a given time interval.
- e) Delta: is the time to checkpoint 80% of aggregate memory of the system.

- f) System Initialization: The time to initialize 99% of the compute resource and 100% of any service resource to the point where a job can be successfully launched, to be defined more specifically dependent on architecture.
- g) System Availability:  $((\text{time in period} - \text{time unavailable due to outages in period}) / (\text{time in period} - \text{time unavailable due to scheduled outages in period})) * 100$

## 9 References

- [1] <http://www.lanl.gov/orgs/hpc/cielo/index.shtml/>, NNSA/ASC Cielo Supercomputer.
- [2] <http://www.nersc.gov/systems/hopper-cray-xe6/>, NERSC Hopper Supercomputer.
- [3] <http://code.google.com/p/portals4/>, Portals 4.0 specification.
- [4] <https://computation.llnl.gov/casc/Pynamic/pynamic.htm/>, Pynamic Benchmark.
- [5] <http://www.cs.sandia.gov/smb/>, Sandia MPI Microbenchmark Suite.
- [6] <https://asc.llnl.gov/sequoia/bnchmarks/>, LLNL Sequoia Benchmarks
- [7] Gary Grider, "Exa-Scale FSIO, Can we get there? Can we afford to?", HEC FSIO 2010 Workshop, Arlington, VA, August 1-4, 2010, LA-UR 10-04611.
- [8] Ning Liu, Jason Cope, Philip Carns, Christopher Carothers, Robert Ross, Gary Grider, Adam Crume, Carlos Maltzahn, "On the role of burst buffers in leadership-class storage systems", IEEE 28<sup>th</sup> Symposium on Mass Storage Systems and Technologies (MSST), April 16-20, 2012.
- [9] John Bent, Sorin Faibish, Jim Ahrens, Gary Grider, John Patchett, Percy Tzeinic, Jon Woodring, "Jitter-free co-processing on a prototype exascale storage stack", IEEE 28<sup>th</sup> Symposium on Mass Storage Systems and Technologies (MSST), April 16-20, 2012.